Aprendizagem Automática
Trabalho Laboratorial – grupos de 1 ou 2 alunos
Rate Beer Dataset

1 Dados

Os dados disponibilizados encontram-se em dois ficheiros `pickle`, um para treino (`rateBeer75Ktrain.p`) com 75000 críticas de cerveja, e outro para teste (`rateBeer25Ktest.p`) com 25000 críticas de cerveja. Os dados foram extraídos do *site* `ratebeer`, que recolheu num período de dez anos críticas de cerveja. Nos ficheiros disponibilizados, cada um contém um dicionário cujo os valores são outros dicionários com a informação referente a uma crítica de uma dada cerveja:

```
beerDic=pickle.load(open('rateBeer75Ktrain.p','rb'))
print(type(beerDic))
dicKeys=list(beerDic.keys())
print(dicKeys[:5])
print(beerDic['000000'])
```

```
<class 'dict'>
['000000', '000001', '000002', '000003', '000004']
{'name': 'Anderson Valley Brother Davids Double',
'style': 'Abbey Dubbel', 'who': 'dmtroyer', 'feel': '3/5', 'look': '3/5', 'smell': '2/5',
'taste': '3/5', 'overall': '4/10', 'review': '22oz.  Pours dark brown with a red rim and
off-white head.  Nose is mostly sweet malt with fairly noticeable alcohol depending on the
temperature.  Flavor is sweet malt with raisin and banana yeast with some slight spice on
the finish.  Very true to the style, but less than inspiring.'}
```

O dicionário referente a cada crítica contém os seguintes campos:

`name`	Nome da cerveja
`style`	Tipo de cerveja
`who`	Nome do crítico
`feel`	Pontuação "tátil" (sobre a sensação): de 1 a 5.
`look`	Pontuação visual: de 1 a 5.
`smell`	Pontuação olfativa: de 1 a 5
`taste`	Pontuação gustativa: de 1 a 5.
`overall`	Pontuação global: de 1 a 10.
`review`	Texto da crítica.

2 Objetivos

Em termos globais, o que se pretende é determinar a qualidade de uma cerveja baseado no que foi escrito sobre a mesma. Neste contexto surgem duas tarefas de classificação a ser implementadas:

I. Classificação Binária:
Nesta tarefa, pretende-se saber se o crítico considera a cerveja muito boa ou muito má, baseado no que escreveu. Considere que uma cerveja é considerada muito boa quando obteve uma pontuação global (campo *overall*) de 9 ou mais valores. Considere ainda que uma cerveja é considerada muito má quando obteve uma pontuação global de 2 ou menos valores.

II. Classificação Multi-Classe:
Prever a pontuação de três aspetos das críticas (*smell, taste* e *overall*). Neste ponto, treine e avalie os classificadores com os dados de treino e verifique se as estimativas do desempenho condizem com os resultados obtidos no conjunto de teste.

3 Desenvolvimento

Deverá ter em conta os seguintes pontos:

1. **Construção do vocabulário**:
   Este ponto é essencial para o desempenho dos modelos usados. Porém, é importante notar que testes exaustivos, tanto a nível da construção do vocabulário como a nível do treino dos modelos implementados, se tornam rapidamente incomportáveis. Assim sendo, é aconselhável para ambas as tarefas de classificação (binária e multi-classe) fazer testes preliminares em que o objetivo não é aferir os melhores modelos para o problema nem encontrar os melhores hiper-parâmetros para os mesmos, mas sim verificar quais as melhores estratégias para a construção do vocabulário. Estas implicam averiguar questões como a limpeza prévia das críticas ou a estimação dos melhores parâmetros da função `TfidfVectorizer`. Neste sentido, é recomendável escolher um único modelo de classificação e ver como as estratégias escolhidas afetam o desempenho do mesmo. Uma vez estas encontradas, é ajuizado encerrar este ponto. Convém igualmente ter em conta que se pretende um vocabulário de menor dimensão possível mas que ao mesmo tempo seja rico suficiente para não afetar o desempenho dos classificadores.

2. **Metodologias de teste e métricas de desempenho**:
   (a) Escolher a metodologia de teste apropriada de modo a ter uma estimação fidedigna do desempenho dos modelos treinados.
   (b) Nos problemas de classificação binária, usar as métricas apropriadas e calibrar os modelos treinados.

3. **Classificadores**:
   (a) Classificação Binária:
      - Grupos individuais: testar dois classificadores.
      - Grupos dois alunos: testar três classificadores.

   (b) Classificação Multi-Classe:
      - Grupos individuais: testar um classificador.
      - Grupos de dois alunos: testar dois classificadores.

4. **Observações Gerais**:
   Deve saber justificar as escolhas feitas no trabalho, tanto a nível da construção do vocabulário, como nas metodologias de treino/testes usadas, e como na seleção dos classificadores implementados. Adicionalmente, deve também fazer uma análise rigorosa dos resultados obtidos.

4 Pontuação

A pontuação pode ser alterada mediante a discussão do projeto.

REQUISITOS MÍNIMOS:
O código deverá estar num ficheiro `.ipynb` (Jupyter Notebook).

10 valores
- Implementação e avaliação dos classificadores para as tarefas de classificação binária e multi-classe. Necessário o bom funcionamento de todos os programas e o cumprimento dos seguintes pontos:
  - Programa(s) de conversão de uma *string* de texto (ou uma listas de *strings*) na representação tf-idf.
  - Programa(s) de treino e de avaliação dos classificadores nas tarefas de classificação multi-classe e binária.
  - O(s) programa(s) de avaliação devem expor claramente os resultados obtidos, preferencialmente através de gráficos ou imagens.

- Apresentação (slides/PowerPoint) com a descrição das experiências efetuadas e dos resultados obtidos.
  - Grupos de 1: Apresentação com o máximo de 20 slides.
  - Grupos de 2: Apresentação com o máximo de 30 slides.

  A apresentação deve ser entregue num ficheiro .pdf com o nome: Axxxxx.pdf (ou AxxxxxAxxxxx.pdf - para grupos de 2 alunos). Tenha em conta a estrutura da apresentação: devem estar claramente identificadas as tarefas abordadas, descritas as experiência efetuadas, métodos usados, resultados obtidos, etc.

VALORES ADICIONAIS:

+ 2 valor Jupyter Notebook: clareza da apresentação, dos comentários e do código.

+ 4 valores Grupos individuais: escolher 1 tópico.
Grupos de 2 alunos: escolher 2 tópicos.

REGRESSÃO Considere que a tarefa de estimar a pontuação da crítica é um problema de regressão. Treine e avalie um modelo de regressão linear. Repita o processo com um modelo de regressão não linear à sua escolha. Compare os resultados da regressão com os obtidos no problema de classificação multi-classe.

PCA Investigue se o pré-processamento dos dados com PCA, é benéfico para o desempenho de um classificador nas tarefas de classificação binária e multi-classe. Determine igualmente qual o número ótimo de componentes principais. Nota: use a função `TruncatedSVD` em vez de `PCA` do sub-módulo `sklearn.decomposition` para poder lidar com matrizes esparsas.

CLUSTERING Use um ou mais algoritmos de *clustering* à sua escolha para agrupar críticas de uma forma não supervisionada. Analise os resultados e indique se os *clusters* estimados fazem sentido. Investigue o efeito da variação do número de *clusters* no desempenho dos algoritmos de agrupamento.

+ 4 valores CONHECIMENTOS ADQUIRIDOS:

Conhecer, saber explicar e como aplicar os seguintes tópicos/métodos no contexto das críticas disponibilizadas.

- Métodos de pré-processamento de dados.
- Métodos de aprendizagem supervisionada.
- Métodos de aprendizagem não supervisionada.
- Calibração e comparação de modelos de classificação binária.

- Metodologias de treino e teste.
- Análise dos resultados obtidos.

BIBLIOTECAS:

Bibliotecas de Python permitidas: `numpy`, `scipy`, `matplotlib`, `sklearn`, `nltk`, `re`, `opencv`, `pickle`, `pandas`, `itertools` e `time`.

-1 valor Casos seja necessário instalar outras bibliotecas, haverá uma penalização de 1 valor.

5 Ficheiros a Entregar e Outros Pontos

-1 valor
- Deve ser entregue, via Moodle, um único ficheiro zip denominado `AxxxxxProjeto.zip` (ou `AxxxxxAxxxxxProjeto.zip` para grupos de 2 alunos).
- O ficheiro zip deverá conter os seguintes ficheiros:
  - **-1 valor** Apresentação: `Axxxxx.pdf` ficheiro pdf com slides de apresentação.
  - **-1 valor** Jupyter Notebook: `Axxxxx.ipynb` (ou `AxxxxxAxxxxx.ipynb` para grupos de 2 alunos).